A COMPARATIVE STUDY OF MACHINE LEARNING ALGORITHMS FOR EEG SIGNAL CLASSIFICATION

Anam Hashmi, Bilal Alam Khan and Omar Farooq

Department of Electronics Engineering, Aligarh Muslim University, Aligarh, India

ABSTRACT

In this paper, different machine learning algorithms such as Linear Discriminant Analysis, Support vector machine (SVM), Multi-layer perceptron, Random forest, K-nearest neighbour, and Autoencoder with SVM have been compared. This comparison was conducted to seek a robust method that would produce good classification accuracy. To this end, a robust method of classifying raw Electroencephalography (EEG) signals associated with imagined movement of the right hand and relaxation state, namely Autoencoder with SVM has been proposed. The EEG dataset used in this research was created by the University of Tubingen, Germany. The best classification accuracy achieved was 70.4% with SVM through feature engineering. However, our prosed method of autoencoder in combination with SVM produced a similar accuracy of 65% without using any feature engineering technique. This research shows that this system of classification of motor movements can be used in a Brain-Computer Interface system (BCI) to mentally control a robotic device or an exoskeleton.

KEYWORDS

EEG. Machine learning. BCI. Motor Imagery signals. Random Forest.

1. INTRODUCTION

Assistive technologies have witnessed tremendous attention and advancements both from the scientific community and industry partners in the last couple of decades. This has led to significant innovation and improvements in the following sector and fields: virtual surgical theatre, robotic surgery, Brain-controlled wheelchairs are the name of the few recent developments. The field of brain-computer interface has also caught the attention of researchers from different fields including neuroscience, cognitive psychology, computer science, and electrical engineering – as it provides the avenue for human welfare and improving life experience. It can be observed inefficient disease diagnosis, development of assistive technologies, health monitoring of the elderly and aiding humanity in general [1]. This study also seeks to explore further this very dimension by analyzing different methodologies used in studying Brain-Computer Interface (BCI). Electroencephalogram or EEG is one of the most common non-invasive methodologies of BCI to record brain signals. It measures the electrical activity of the brain using electrodes that are placed over the scalp. EEG is preferred because of its ease of portability and capturing high temporal brain information, however, it fails in capturing high spatial information [2]. BCI uses these EEG signals associated with the user's activity and then apply different signal processing algorithms for translating the recorded signals into control commands for different applications. In an EEG there are five types of oscillatory waves that are commonly used for analysis, which are:

(a) delta (0.5–4 Hz);
(b) theta (4–7 Hz);

DOI: 10.5121/sipij.2021.12603

(c) alphaormu (7–13 Hz);
(d) beta (13–25 Hz);
(e) gamma (25–50 Hz).

Motor imagery (MI) is a process in which an individual rehearses or stimulates an action. It is a very popular paradigm in the analysis of an EEG based BCI system. MI activity usually lies in alpha (or mu) and beta bands [3].

In the past few years, significant advances have been made in the BCI systems and they have revolutionized rehabilitation engineering by providing differently-abled individuals with a new avenue to communicate with the external environment. According to many works of literature, the strength of a BCI system depends upon the methods in which the brain signals are translated into control commands of machines. A novel method namely an arc detection algorithm to find an optimal channel was proposed by ErdemEkran and Ismail Kurnaz [4]. For feature extraction DWT was used and several machine learning algorithms were used for classification purposes, which were SVM, K- nearest neighbour, and Linear Discriminant Analysis. The best accuracy achieved by their methodology was 95% in classifying ECoG signals (BCI competition III, dataset I). Jun Wang and Yan Zhao proposed feature selection based on one dimension real-valued particle swarm optimization, extracted nonlinear features such as Approximate entropy and Wavelet packet decomposition, and achieved the best accuracy of 100% [5]. Khan B. A. et al. (2020), employed feature engineering and linear discriminant analysis to accurately classify EEG signals associated with the seizure. The authors have suggested the use of novel features such as Gini’s coefficient to extract features and employed a very simple Linear discriminant analysis based method to accurately classify seizure data with an accuracy of 100% [16]. In another work, the authors' Khan B. A. et al., have used very simple statistical features to classify imagined and executed hand movements using EEG signals [17]. Aswinseshadri. K et al. used the wavelet packet tree for feature extraction. They used a genetic algorithm, applied information gain, and mutual information to find the best feature set and for classification K-NN and Naïve Bayes were employed [6]. Chea-Yau Kee et al. proposed a novel feature known as Renyi entropy that has been employed for feature extraction and BLDA for classification [7]. K. Venkatachalam et al. proposed the use of the Hybrid-KELM (Kernel Extreme Learning Machine) method based on PCA (Principal Component Analysis) and FLD (Ficher’s Linear Discriminant) analysis for MI BCI classification of EEG signals. The best accuracy reported was 96.54% [8]. Rajdeep Chatterjee et al. used the AAR (Auto Adaptive Regressive) algorithm for feature extraction, proposed a novel feature selection method based on IoMT (Internet of Medical Things), and classified EEG signals using SVM and ensemble variants of classifiers. The best accuracy reported was 80% [9]. The authors of [10] employed a combination of common spatial patterns (CSP) and local characteristic-scale decomposition (LCD) algorithm for feature extraction, a combination of firefly algorithm and learning automata (LA) to optimize feature selection, and spectral regression discriminant analysis (SRDA) classifier for classifying MI-EEG signals. They have used this method for a real-time brain-computer interface to show their method’s efficiency.

Several studies have usually worked on the classification of right vs left-hand movement, or hand vs tongue movements, or hands vs legs movements. There is very limited literature that has studied and classified intricate hand movements such as opening and closing of a hand, or movements of different fingers, or classification of different hand gestures using neural signals, and those who have worked on these subjects either did not achieve high enough accuracy or failed to work in a real-world setup. This paper probed this very aspect of studying intricate human motions and worked on the classification of imagining of opening and relaxing of a hand using MI-EEG signals.

The contributions of this paper are following:

(i) Comparison of different machine learning algorithms to empirically establish a proper method that could provide satisfactory results for this dataset.

(ii) Accurate classification of the motor imagery signals using a deep learning-based algorithm (Autoencoder) which utilizes raw EEG signals and sends them for classification. Since the pipeline is made independent of this dataset, it should work for other similar physiological signal datasets.

The organization of this paper is as follows: the first part is the Introduction stage, where a brief introduction was provided and related works were reported, followed by the Materials and Methodology stage. In this part, the materials or data that was used in the paper is described and the methodology of this work was elucidated. The third stage involves the results of the study with a detailed discussion and a conclusion along with the limitation of this study and future scope.

2. MATERIALS AND METHODOLOGY

2.1. Data Used

The data used in this study was taken from [11]. The data consist of EEG recordings of a single subject. The subject was connected with a high spinal cord lesion and was controlling an exoskeleton (Brain-Neural computer interface) attached to his paralysed limb. The cue-based BNCI paradigm consisted of two different tasks, namely the ‘imagination of movement’ of the right hand (Class 1) and ‘relaxation/no movement’ (Class 2).

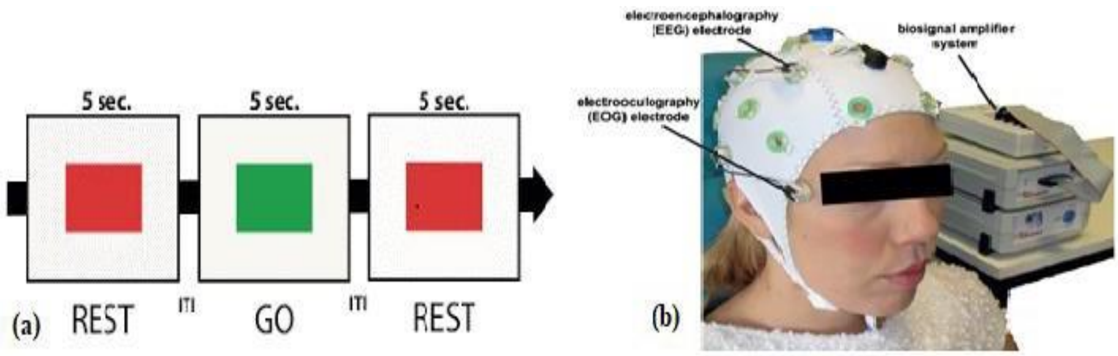


Figure 1. Timing Scheme of the trials used (a) and a subject during the EEG recording of the Dataset (b).

A randomly shown visual cue is used to indicate to the user when to open (for Green square) and when to close (for Red Square). These two indications were given 24 times each in total separated by inter-trial intervals (ITIs) of 4-6 seconds. Each indication was displayed for 5 seconds after which the device was driven back to the open position. Re-setting the exoskeleton into open position required one second.

EEG was recorded from 5 conventional EEG recording sites F4, T8, C4, Cz, and P4 according to the international 10/20 system using an active electrode EEG system (Acti-cap® and BrainAmp®, BrainProducts GmbH, Gilching, Germany) with a reference electrode placed at FCz and ground electrode at AFz. EEG was recorded at a sampling rate of 200 Hz, bandpass filtered at 0.4-70Hz and pre-processed using a small Laplacian filter.

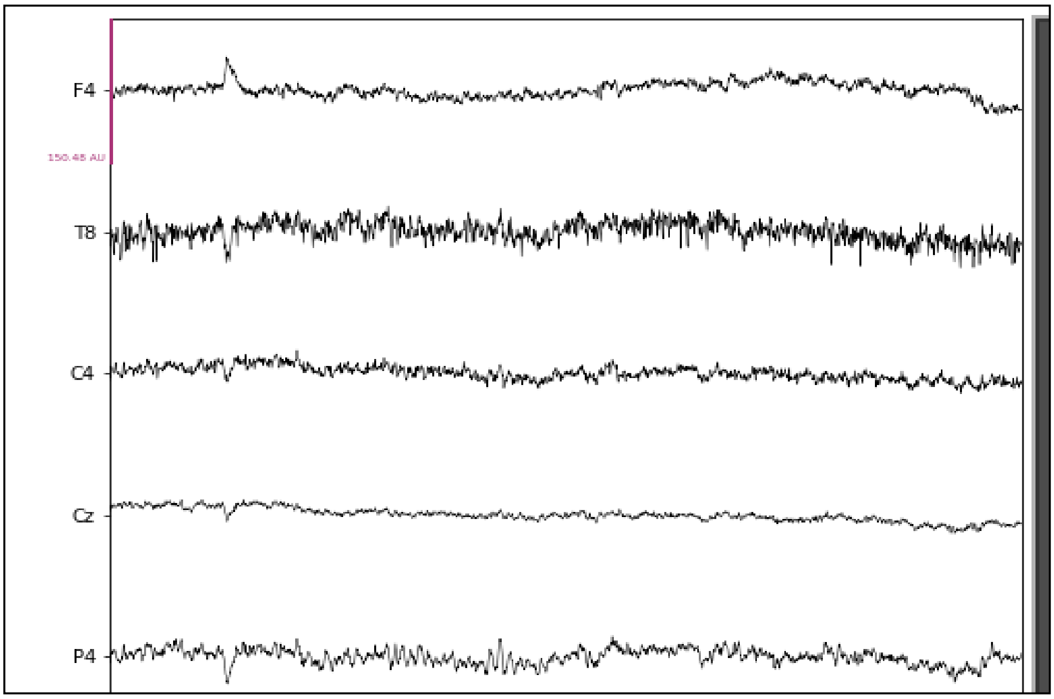
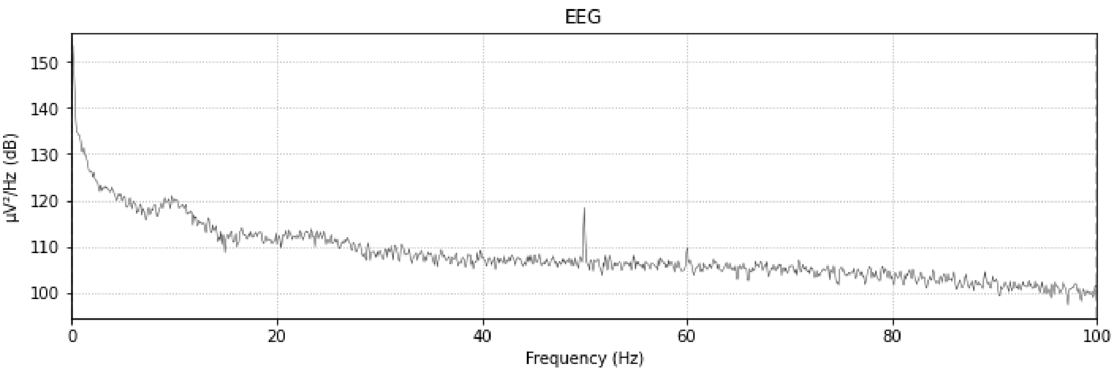


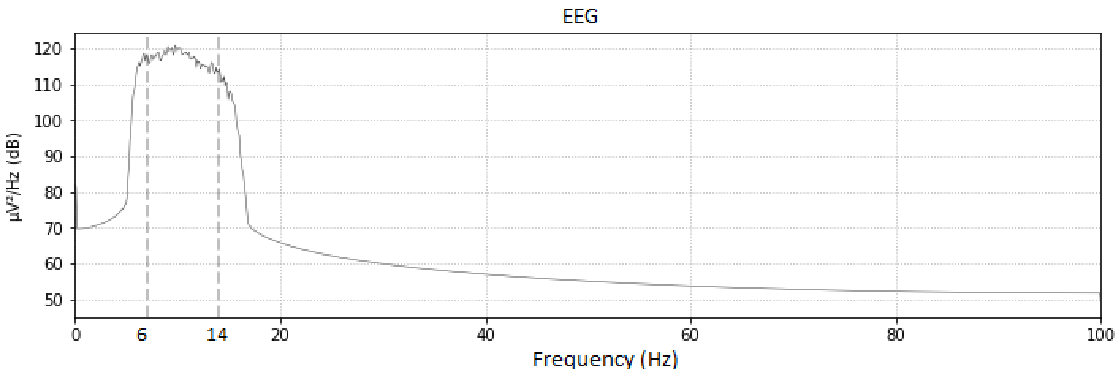
Figure 2. Showing raw EEG channel waveforms of different channels [11].

2.2. Pre-Processing

At this stage, the data was processed or filtered to capture information related to Motor Imagery. While the recording of EEG stores different noise elements from line frequencyinterference to different unwanted artefact signals. All of which could frustrate the model's classification progress. Therefore, it is necessary to first remove these noise elements and unwanted signals before the actual analysis. Many electrophysiological features are associated with the brain's normal motor output channels [12]. Several studies have suggested the presence of mu rhythm in the frequency range of 7-13 Hz. Some of these important features are the mu (8-12 Hz) and beta (13-30 Hz) rhythms [13]. Therefore, this particular band has been selected for further analysis. To this end, an FIR filter was designed for the frequency range of 6 to 14 Hz with Hamming window of 0.0194 passband ripple and 53 dB stopband attenuation.



(a)



(b)

Figure 3. Showing power spectral density of EEG signal channel 'C4' (a) before filtering and (b) after filtering using the designed FIR filter.

2.3. Feature Extraction

A feature is a measurable property or characteristic of an observed signal. It should be informative, discriminative and orthogonal to other features. Feature extraction is the method of extracting these features. It can be defined as the process of transforming original data into a dataset with a reduced number of variables but with the most discriminative information.

After the pre-processing stage, the channels were selected from the F4, T8, C4, Cz and P4 based on literature review and using correlation-based analysis. Finally, Cz and P4 channels were selected as they were amongst the most commonly used channels for motor imager classification purposes [17]. To compare different machine learning algorithms certain features were extracted. The choice of these features was based on the results from our previous work [19] and other commonly used features. These features include Mean absolute value (MAV), Variance, Median Absolute Deviation (MAD), Variance, Energy, Spectral Entropy, and Mean. Particularly, the two classes – which corresponds to imagining of the opening of hand as 'class 1' and relaxation or no movement as 'class 2' – differ in dispersion. The same can be observed from the histogram of class 1 and class 2 appears, where the class 1 histogram appears to be skewed from the normal distribution. Thereby justifying the choice of IQR, MAD, Variance, Standard Deviation, Skewness and Kurtosis. Energy and MAV were chosen because it has been reported in many works that mu rhythm has a lower amplitude than that of the alpha wave [14].

The following are the mathematical equations of the extracted features:

2.3.1. Mean Absolute Value (MAV)

It is defined as the mean value of the absolute values of the data. Mathematically,

$$MAV = \frac{1}{N}\sum_{i=1}^{N}|X_i(n)| \qquad \text{(i)}$$

2.3.2. Variance

It is defined as the expectations of the squared deviation of a random variable from its mean.

$$Var(X) = E[(X-\mu)^2] \qquad \text{(ii)}$$

Where Var(X) computes of variance of data X, 'μ' represents the average value, 'E' represents the expectation.

2.3.3. Median Absolute Deviation

It is defined, as the name suggests, as the median value of the absolute deviations from the data median value.

$$MAD = median(|Xi - median(X)|) \qquad \text{(iii)}$$

Where Xi is the ith value of the data X.

2.3.4. Spectral Entropy

The spectral entropy (SE) of a signal is a measure of its spectral power distribution. X(m) is the discrete Fourier transform of the signal x(n). S(m) is the power spectrum of the X(m). P(m) is the probability distribution of S(m) and H is the Spectral Entropy, calculated based on the Shannon entropy formula.

$$S(m) = |X(m)^2|$$

$$P(m) = \frac{S(m)}{\sum_i S(i)}$$

$$H = -\sum_{m=1}^{N} P(m) log_2 P(m) \qquad \text{(iv)}$$

2.3.5. Energy

It is the area under the squared magnitude of the considered signal. Mathematically,

$$Es = \sum_{n=-\infty}^{\infty} |X(n)|^2 \qquad \text{(v)}$$

2.3.6. Mean

It is defined as the averaged sum of a series of numbers. It can be calculated as,

$$Mean = \frac{\sum_i x_i}{n} \qquad \text{(vi)}$$

2.4. Methodology

The study followed a very simple pipeline –starting from the pre-processing of unwanted artefacts and channel selection, then feature were extracted, and finally, different combinations of features were classified using different machine learning algorithms to empirically observe which algorithm would work best for this task.

In this study, two classes are corresponding to the Motor Imagery (MI) tasks; hand opens and hand relaxes. The data was pre-processed and filtered using an FIR bandpass filter. This band was considered as it corresponds to the Mu rhythm (7 – 13 Hz) where the motor activity in the brain is usually associated [khan B. A.]. The total duration of 'Class 1' is 45 seconds and the sample rate is 200 Hz, which produces 9000 data points.

One of the pipelines used in this study involves the use of traditional feature engineering steps and employing a classifier to do classification. To this end, the following steps have been followed. A one-second sliding window is considered for the analysis of the signal and each second 200 samples are considered, for which 6 features were extracted which produce a feature vector of [45×1] corresponding to each feature. This process was performed until the end of the recording, thereby producing a feature vector of the size of [270×1]. This was repeated for the selected channels (Cz and P4) producing a feature matrix of [270×2]. After the feature extraction, different combinations of these features were considered. At each selected feature combination different training and testing sizes were split to see classification performance across different training and test split sizes. This has also provided insight into algorithms that are relatively more robust to such changes. Table 1. Shows different feature combinations considered in this study.

The other pipeline – based on autoencoder with other classifiers – employed in this study does not make use of traditional feature engineering steps and skips over this. It directly utilizes raw EEG signals followed by the classification stage using SVM, Random Forest, and K-nearest neighbour. Details about these classifiers and used methodology have been described below:

Table 1. showing features and their combination.

Feature name	**Feature code**	**Feature combination**
Mean absolute value	F1	F5
Variance	F2	F5+F6
Median absolute deviation	F3	F3+F5+F6
Spectral Entropy	F4	F1+F3+F5+F6
Energy	F5	F1+F2+F3+F5+F6
Mean	F6	F1+F2+F3+F4+F5+F6

Here, Autoencoder will be described as the other methodology is defined above. Autoencoders are feedforward neural networks that can learn efficient representations of the input data without the need for labels in the training data. Autoencoders are regarded as powerful feature extractors. So, autoencoders work by learning efficient ways to represent the input data by copying their inputs to their outputs. In the learning process of the autoencoders, we can put several constraints on the way these learn the internal representations of the input data, such as reducing the number of features, which will make the autoencoder work as dimensionality reduction networks. The architecture of the autoencoder we used consisted of one input layer, one hidden layer and one output layer. The number of nodes used in the hidden layer was 5. The autoencoder network architecture consists of an encoder part and a decoder part, the encoder part is responsible for coding the input data while the decoder part reconstructs the input from the code [18]. The activation function we used in the autoencoder network was Exponential Linear Unit (ELU). The reason for choosing the ELU activation function was that it is a non-saturating function and thus, doesn’t suffer from the vanishing gradients problem. After encoding and decoding raw EEG signals, the decoded signals are sent for classification using the best classifiers observed by using classification accuracy as the metrics. These classifiers included – SVM, Random forest, and K-nearest neighbour using Sklearn library of python [20]. Their corresponding accuracies are reported in the result section.

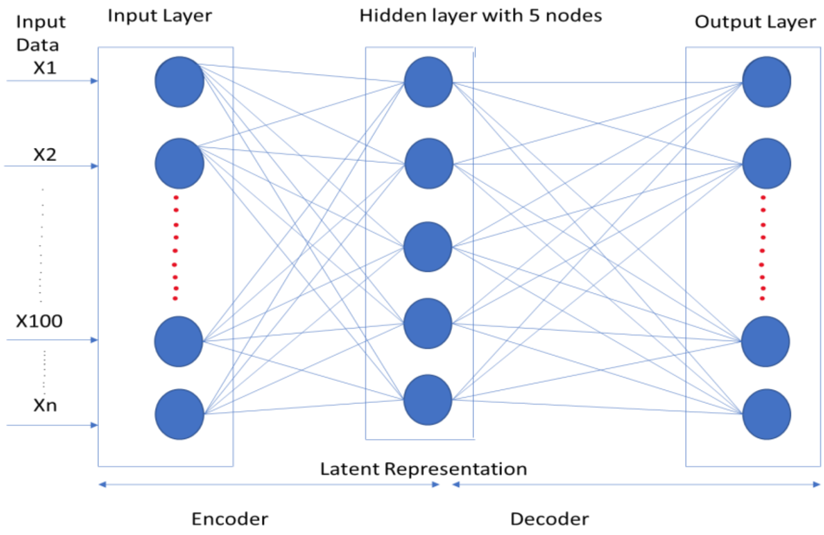


Figure 4. Autoencoder depiction with input layer, hidden layer and output layer. In this study, input layer has the dimension of input data, hidden layer has only 5 nodes and output layer is again of the same size as input.

K-nearest neighbour (KNN)

The k-nearest neighbours (KNN) algorithm is a non-parametric, supervised machine learning algorithm, that is both simple and powerful. The KNN algorithm works by assuming that the data points that are similar exists in close proximity to each other and works on the idea of a similarity function (example, distance functions like Euclidean).

This method is used independently as well as along with the autoencoder pipeline. For the KNN classifier, the values of hyperparameters chosen were: the number of neighbours to use (n_neighbors) was taken as 5 and the distance metric we chose was 'Euclidean' distance.

Support Vector Machine (SVM)

One of the most powerful and versatile machine learning algorithms is Support Vector Machine (SVM), which can perform both linear and non-linear classification. The Support vector machines work by finding a hyperplane which is a decision boundary in N-dimensional space (N-the dimensions of feature space) for distinctly classifying the data points. The objective of the SVM is to generate a maximal marginal hyperplane that can divide the dataset into distinct classes in the best possible way. Next, we used Support Vector Machines using the Sklearn library of python [20].

For the SVM classifier, the values of hyperparameters chosen were 0.1 for the regularization parameter C, the kernel used was 'radial basis function' ('rbf') and the value for the kernel coefficient 'gamma' used was 1.

Random Forest Classifier

Random Forests are known as ensemble learning classifiers and usually gives good results without much hyperparameter tuning. These work by constructing a number of decision trees during training by using the split criteria for the decision nodes as 'Gini' impurity or 'entropy, and the output of which is chosen as the class most selected by the decision trees.Finally, we used the Random Forest Classifier using the Sklearn library of python [20].

For the Random Forest classifier, the values of hyperparameters chosen were 300 for the number of trees in the forest (n_estimators), the function to measure the quality of split of the nodes of the decision was 'Gini' impurity.

Multi-layer Perceptron (MLP)

The perceptron is a single neuron while the Multi-layer Perceptron (MLP) is a class of artificial neural networks that uses supervised learning and are composed of multiple layers of the perceptron. Multi-layer perceptron can classify data that is not linearly separable unlike that of a perceptron and consists of at least three layers, the input layer, the hidden layer, and an output layer.

Linear Discriminant Classifier (LDA)

Linear Discriminant Classifier (LDA) is a very simple supervised classification algorithm that works by finding a combination of linear features to separate the data into two or more classes. LDA considers two assumptions for the classification task, one is that it assumes that the data has Gaussian distribution and the second is that the classes in the dataset have the same covariance matrices.

3. CLASSIFICATION

The classification of EEG signals plays a vital role in biomedical research. According to [15], there are mainly 5 types of classifiers used in BCI research such as linear classifiers, nonlinearclassifiers, neural networks, nearest neighbour classifiers and a combination of these. In this study, all of these classifiers have been compared to empirically establish which classifier would be most appropriate for this "task". These include Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), K-Nearest Neighbour (KNN), Random Forest (RF), Multi-layer Perceptron (MLP) and the combination of these classifiers using majority voting criteria. The best three classifiers were selected based on the classification accuracy to accurately classify motor imagery signals.

Finally, a deep learning-based autoencoder along with SVM was used to classify imagined movement of hand using different classifiers including SVM, Random Forest, and K-nearest neighbour. EEG signals were directly input to the autoencoder which encodes and decode raw EEG and performs dimensional reduction. The decoded output will be used in the classification stage and will be classified using these different mentioned classifiers.

4. RESULTS

Different machine learning algorithms have been used in this study by comparing their classification accuracy and their robustness by changing parameters such as feature combination and train-test data size. Results from these changes have been reported here below from Table 2 to Table10. We have used classification accuracy in order to evaluate the effectiveness of our method.

$$Classification\ accuracy\ (\%) = \frac{TP+TN}{TP+FP+FN+TN} \qquad \text{(i)}$$

Where,

TP is True Positive;
TN is True Negative;
FP is False Positive;
FN is False Negative

Table 2 summarises results from Table 3 to Table 10. It provides a brief overview of how different classifiers have been used and which provided the best accuracy. Moreover, it also shows the combination of classifiers that have provided the best accuracy based on majority voting criteria. Table 2a shows a different combination of classifiers. For combining, classifiers are so chosen based on their classification accuracy. The top three classifiers have been chosen for each feature combination. Finally, only that classifier combination is reported which yielded the best performance. Table 3 onwards reports results for classification using different feature combinations – as has already been shown in Table 1 – by different classifiers and by their combination as well. As can be seen in Table 3, shows results for different feature combinations and corresponding classifier accuracies. Based on this table, the three best classifier combinations are selected and whichever combination yields higher accuracy were reported. These best combination accuracies have been reported in Table 4.

Table 2. Summary table of best classification accuracy corresponding to different feature combination.

S.no	Feature combination	Best train-test split	Best classifiers	Best combination accuracy
1	F1+F2+F3+F4+F5+F6	85-15	**SVM - 70.4**	**63 (RF+SVM+KNN)**
2	F1+F2+F3+F5+F6	85-15	**SVM- 66.67**	**66.67 (RF+SVM+KNN)**
3	F1+F3+F5+F6	85-15	**SVM- 66.67**	**63 (RF+SVM+KNN)**
4	F3+F5+F6	85-15	**SVM- 70.4**	**63 (MLP+SVM+KNN)**
5	F5+F6	85-15	**MLP - 66.67**	**63 (MLP+SVM+KNN)**
6	F5	85-15	**SVM- 70.4**	**70.4 (MLP+SVM+KNN)**

Table 2a. Showing different combination of classifiers utilizing majority voting criterion for improving their performance.

S. No.	Combination of Classifier	Acronym
1	Random Forest + Support Vector Machine + K-nearest neighbour : (RF+SVM+KNN)	T1
2	Multi-layer Perceptron + Support Vector Machine + K-nearest neighbour : (MLP+SVM+KNN)	T2
3	Linear Discriminant Analysis+ Random Forest + Support Vector Machine + Multi-layer Perceptron + K-nearest neighbour (LDA+RF+SVM+MLP+KNN)	T3

Table 3. Classification accuracies of different classifiers for channels ‘Cz’ and ‘P4’ for different feature combination.

S. No.	Train-Test split	Features	LDA	RF	MLP	KNN	SVM
1	85-15	F1+F2+F3+F4+F5+F6	44.45	55.6	44.45	55.6	70.4
2	85-15	F1+F2+F3+F5+F6	51.9	63	44.45	55.56	66.67
3	85-15	F1+F3+F5+F6	48.14	51.8	44.45	44.45	66.67
4	85-15	F3+F5+F6	40.7	48.14	59.25	51.85	70.37
5	85-15	F5+F6	44.45	55.56	66.67	62.9	62.9
6	85-15	F5	29.6	48.14	66.67	59.25	70.37

Table 4. Showing best classification accuracy by combining best three classifiers using majority criterion for different combination of features.

Train-Test split	Features	Best three classifiers	Best Classification accuracy using majority voting criteria
85-15	F1+F2+F3+F4+F5+F6	T1: MLP+SVM+KNN	**62.96**
85-15	F1+F2+F3+F5+F6	T2: RF+SVM+KNN	**66.7**
85-15	F1+F3+F5+F6	RF+SVM+KNN	**62.96**
85-15	F3+F5+F6	MLP+SVM+KNN	**62.96**
85-15	F5+F6	MLP+SVM+KNN	**62.96**
85-15	F5	MLP+SVM+KNN	**70.37**

Table 5. Showing results from a six selected combination of features for different training and test data split.

Train-Test Split	Feature combination selected	LDA	RF	MLP	SVM	KNN	Best combined
80-20	F1+F2+F3+F4+F5+F6	41.67	61.11	47.22	55.56	58.33	**61.11 (T1)**
70-30	F1+F2+F3+F4+F5+F6	46.3	55.56	44.44	50	55.56	**57.4 (T3)**
60-40	F1+F2+F3+F4+F5+F6	50	56.9	40.3	40.3	47.22	**47.22 (T1)**
50-50	F1+F2+F3+F4+F5+F6	46.7	52.22	47.8	57.8	47.8	**50 (T1)**

Table 6. Showing results from a five selected combination of features for different training and test data split.

Train-Test Split	Feature combination selected	LDA	RF	MLP	SVM	KNN	Best combined
80-20	F1+F2+F3+F5+F6	47.22	58.33	55.56	55.56	55.56	**63.88 (T1)**
70-30	F1+F2+F3+F5+F6	51.85	53.7	44.44	57.4	53.7	**55.56 (T1)**
60-40	F1+F2+F3+F5+F6	54.16	52.8	40.3	54.2	55.56	**55.56 (T3)**
50-50	F1+F2+F3+F5+F6	51.11	53.33	47.8	57.8	60	**58.9 (T1)**

Table 7. Showing results from a four selected combination of features for different training and test data split.

Train-Test Split	Feature combination selected	LDA	RF	MLP	SVM	KNN	Best combined
80-20	F1+F3+F5+F6	44.44	55.56	50	55.56	47.22	**58.33 (T1)**
70-30	F1+F3+F5+F6	46.9	57.4	44.44	61.11	51.85	**59.3 (T2)**
60-40	F1+F3+F5+F6	47.22	52.78	40.3	56.9	51.3	**52.8 (T1)**
50-50	F1+F3+F5+F6	45.56	51.11	47.8	55.56	52.22	**52.2 (T1)**

Table 8. Showing results from a three selected combination of features for different training and test data split.

Train-Test Split	Feature combination selected	LDA	RF	MLP	SVM	KNN	Best combined
80-20	F3+F5+F6	50.0	55.56	66.67	51.11	55.56	**61.11 (T2)**
70-30	F3+F5+F6	42.59	48.14	44.44	68.52	55.56	**61.11 (T2)**

60-40	F3+F5+F6	41.67	51.4	40.3	66.67	55.56	**61.11 (T2)**
50-50	F3+F5+F6	40	56.67	47.78	61.11	55.56	**58.89 (T1)**

Table 9. Showing results from a two selected combination of features for different training and test data split.

Train-Test Split	Feature combination selected	LDA	RF	MLP	SVM	KNN	Best combined
80-20	F5+F6	44.44	58.33	63.88	58.33	55.56	**58.23 (T1)**
70-30	F5+F6	42.22	48.89	62.22	57.80	55.6	**62.22 (T2)**
60-40	F5+F6	36.11	51.38	45.83	48.61	47.22	**50.0 (T1)**
50-50	F5+F6	41.11	57.78	41.12	56.67	54.45	**60.0 (T1)**

Table 10. Showing results from a single for different training and test data split.

Train-Test Split	Feature combination selected	LDA	RF	MLP	SVM	KNN	Best combined
80-20	F5	30.55	52.78	66.67	66.67	58.33	**66.67 (T2)**
70-30	F5	29.62	46.29	55.6	53.7	51.8	**54.67 (T2)**
60-40	F5	26.38	47.22	36.11	48.61	47.22	**47.22 (T1)**
50-50	F5	32.23	53.34	52.22	53.33	56.67	**57.78 (T1)**

Table 11. showing results from Autoencoder based method of classification.

S. No.	**Feature**	**Classifier**	**Accuracy (%)**
1	Autoencoder	KNN	58
2	Autoencoder	SVM	65
3	Autoencoder	Random forest	63

5. Discussion and Future work

In this study, different statistical features such as Mean Absolute Value, Median Absolute Deviation, Variance, Spectral Entropy, Mean, and Energy were used to extract the underlying information from a dynamic EEG. This study has shown that the proposed features were successful in capturing the relevant distinguishing information. This study has also compared different machine learning algorithms with one another under different conditions to observe the method's robustness to small changes. From the results, it can be observed that SVM has produced consistently decent accuracy and was very least affected by changes in different feature combinations or changes in train-test data split. After SVM, KNN and Random Forest were two other algorithms that have shown significant promise in that aspect. On the other hand, Linear Discriminant analysis has consistently performed very poorly. This is something that was expected as the data is not linearly separable, which means linear classifiers would not be able to work well on the dataset. This would also explain the inconsistent performance of MLP. This can also be observed from the best classifier combination – most of the classifier combination does not include a linear classifier. The summary table accurately summarizes this aspect in showing the classifier and method that have been robust and also produced decent classification accuracies.The important thing to note here is that the SVM has been reported in this study as the robust method for the classification of EEG signals. However, this does not mean that its performance was indifferent to the changes in the train-test data split. Its performance does get affected – it decreases with increasing train-test data split – but is relatively stable when compared with other classifiers. The best accuracy has also been achieved with SVM based method of 70%. In our previous work, Hashmi A. et al. 2021, the best accuracy of 85% was achieved after carefully cleaning the signals, applying pre-processing steps, extracting desired frequency range through extremely computationally expensive method – wavelet decomposition method, carefullyextracting relevant features and then ranking those features with the help of random forest after which classification process was performed [19]. In this study, we tried to reduce these computational expensive methods and replaced them with other very simple methods such as using a simple FIR bandpass filter instead of wavelet decomposition to gain the desired frequency range. Even after so many additional processing steps, the overall average accuracy was around 75% which is quite similar to what this study has achieved. Also, this study has developed another method – autoencoder with SVM. In this method, raw EEG signals will be directly input in the autoencoder which will encode and decode it – reducing the dimensionality. The decoded output will go into the SVM classifier which has been classified into two classes. The average accuracy in this method was 65% and this method was very much indifferent to any type of changes in the train-test data split. This method was developed without considering any of the specificities of this dataset, so theoretically this method should be able to work decently on other physiological datasets. The other classifier used with the autoencoder method were K-nearest neighbour and Random forest. The corresponding accuracies for these methods were 58% and 63%. It is worthwhile to mention here that these accuracies were not affected at all by the change in train-test data split, meaning this method is quite robust for future use. However, there is a need to improve these accuracy results by doing some sort of pre-processing of data. This can be further explored in future studies – how much pre-processing would be necessary to make a significant increase in classification accuracy? The other limitation of this study is that the data, although very real-world and relevant, was very limited. This can be frustrating if deep learning-based methods are to be employed. With these limitations, this study has tried to empirically observe the effect of certain parameters such as train-test data split and features on the classification accuracy and has compared different machine learning-based algorithms against each other. A new robust approach based on deep learning-based methods has also been proposed along with the comparison with our previous work.

The application of this method in the future is that it can be used to control an external device i.e. Neuro-prosthetics. The translated commands will be used as input to the external device via a computer (or micro-controller). This will, in turn, provide basic operations of the device. This study could also be used in the supervision of a trained physiotherapist to provide functional restoration to patients with spinal cord injury. In addition to that, this method can also be used in sports Biomechanics.

REFERENCES

[1] S. R. Guntur, R. R. Gorrepati, and V. R. Dirisala, "Robotics in Healthcare : An Internet of Medical Robotic Things (IoMRT)" Perspective. Elsevier Inc., 2016.

[2] M. Radman and A. Chaibakhsh, “Generalized Sequential Forward Selection Method for Channel Selection in EEG Signals for Classification of Left or Right Hand Movement in BCI,” ICCKE, pp. 137–142, 2019.

[3] N. E. Isa, A. Amir, M. Z. Ilyas, and M. S. Razalli, “Motor imagery classification in Brain- computer interface (BCI) based on EEG signal by using machine learning technique”, Bulletin of Electrical Engineering and Informatics, vol. 8, no. 1, pp. 269–275, 2019.

[4] E. Erkan and I. Kurnaz, “A study on the effect of psychophysiological signal features on classification methods,” Measurement, vol. 101, pp. 45–52, 2017.

[5] J. Wang, “EEG signal classification with feature selection based on one-dimension real- valued particle swarm optimization,” International Conference on Mechatronics, Control and Electronic Engineering, pp. 310–314, 2014.

[6] K. Aswinseshadri and V. T. Bai, “Feature Selection In Brain-Computer Interface Using Genetics Method,” 2015 IEEE Int. Conf. Comput. Inf. Technol. Ubiquitous Comput. Commun. Dependable, Auton. Secur. Comput. Pervasive Intell. Comput., pp. 270–275, 2015.

[7] C. K. S. G. Ponnambalam, “Binary and multi-class motor imagery using Renyi entropy for feature extraction,” Neural Comput. Appl., vol. 28, no. 8, pp. 2051–2062, 2017.

[8] K. Venkatachalam, A. Devipriya, J. Maniraj, M. Sivaram, A. Ambikapathy, and I. S. Amiri, “A novel method of motor imagery classification using EEG signal”, Artificial Intelligence In Medicine, vol. 103, no. December 2019, 2020.

[9] R. Chatterjee, T. Maitra, S. K. Hafizul, and M. Mehedi, “A novel machine learning-based feature selection for motor imagery EEG signal classification on the Internet of medical things environment,” Futur. Gener. Comput. Syst., vol. 98, pp. 419–434, 2019.

[10] Liu, A., Chen, K., Liu, Q., Ai, Q., Xie, Y., Chen, A.: Feature Selection for Motor Imagery EEG Classification Based on Firefly Algorithm and Learning Automata. Sensors. 17, 2576 (2017).

[11] M. Witkowski, M. Cortese, M. Cempini, J. Mellinger, and N. Vitiello, “Enhancing brain- machine interface (BMI) control of a hand exoskeleton using electrooculography (EOG)”, Journal of Neuro Engineering and Rehabilitation, pp. 1–6, 2014.

[12] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, “Brain-computer interfaces for communication and control,” Clin. Neurophysiol., vol. 113, no. 6, pp. 767–791, 2002.

[13] D. J. McFarland, L. a Miner, T. M. Vaughan, and J. R. Wolpaw, “Mu and beta rhythm topographies during motor imagery and actual movements.,” Brain Topogr., vol. 12, no. 3, pp. 177–186, 2000.

[14] G. Pfurtscheller, A. Stancák, and C. Neuper, “Event-related synchronization (ERS) in the alpha band — an electrophysiological correlate of cortical idling: A review,” Int. J. Psychophysiol., vol. 24, no. 1, pp. 39–46, 1996.

[15] H. Bashashati, R. K. Ward, G. E. Birch, and A. Bashashati, “Comparing Different Classifiers in Sensory Motor Brain-Computer Interfaces”, PLOS, pp. 1–17, 2015.

[16] Khan, Bilal Alam, Anam Hashmi, and Omar Farooq. "An Automated System for Epileptic Seizure Detection Using EEG." In Advances in Data and Information Sciences, pp. 147-155. Springer, Singapore, 2020.

[17] B. A. Khan, A. R. Usmani, S. Athar, A. Hashmi, O. Farooq, and M. Muzammil. "EEG-Based Exoskeleton for Rehabilitation Therapy." In Ergonomics for Improved Productivity, pp. 645-653. Springer, Singapore, 2021.

[18] P. K. Parashiva and A. P. Vinod, "A New Channel Selection Method using Autoencoder for Motor Imagery based Brain Computer Interface," 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), 2019, pp. 3641-3646, doi: 10.1109/SMC.2019.8914251.

[19] A. Hashmi, B. A. Khan, O. Farooq, “A study of the classification of Motor imagery signals using Machine learning tools,” David C. Wyld et al. (Eds): VLSIA, INWES, CSIP, BMLI - 2021 pp. 23-33, 2021. CS & IT - CSCP 2021, doi: 10.5121/csit.2021.112104

[20] Pedregosa et al., “Scikit-learn: Machine Learning in Python,”Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.